Regular Article

2D Parity Product Code for TSV Online Fault Correction and Detection

Khanh N. Dang[1], Michael Conrad Meyer[2], Akram Ben Ahmed[3], Abderazek Ben Abdallah[4], Xuan-Tu Tran[1]

[1] VNU Key Laboratory for Smart Integrated Systems (SISLAB), University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[2] G.S. of Information, Production and Systems, Waseda University, Kitakyushu, Japan
[3] National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
[4] Adaptive Systems Laboratory, The University of Aizu, Aizu-Wakamatsu, Fukushima, Japan

Correspondence: Khanh N. Dang, khanh.n.dang@vnu.edu.vn
Communication: received 13 July 2019, revised 16 October 2019, accepted 28 October 2019
Online publication: 4 March 2020, Digital Object Identifier: 10.21553/rev-jec.242
The associate editor coordinating the review of this article and recommending it for publication was Dr. Tran Thi Hong.

Abstract– **Through-Silicon-Via (TSV) is one of the most promising technologies to realize 3D Integrated Circuits (3D-ICs). However, the reliability issues due to the low yield rates and the sensitivity to thermal hotspots and stress issues are preventing TSV-based 3D-ICs from being widely and efficiently used. To enhance the reliability of TSV connections, using error correction code to detect and correct faults automatically has been demonstrated as a viable solution. This paper presents a 2D Parity Product Code (2D-PPC) for TSV fault-tolerance with the ability to correct one fault and detect, at least, two faults. In an implementation of 64-bit data and 81-bit code-word, 2D-PPC can detect over 71 faults, on average. Its encoder and decoder decrease the overall latency by 38.33% when compared to the Single Error Correction Double Error Detection code. In addition to the high detection rates, the encoder can detect 100% of its gate failures, and the decoder can detect and correct around 40% of its individual gate failures. The squared 2D-PPC could be extended using orthogonal Latin square to support extra bit correction.**

Keywords– **Fault-Tolerance, Error Correction Code, Through Silicon Via, Product Code, Parity.**

1 Introduction

Through-Silicon-Vias (TSVs) serve as vertical wires between two adjacent layers in Three Dimensional Integrated Circuits (3D-ICs). Thanks to their extremely short lengths, their latency are low which could offer extremely high speeds of communication [1, 2]. In fact, the authors in [2] presented a 20 GHz TSV model that offers up to 10 Gbps input signals. Moreover, as a 3D-IC technology, TSV-based ICs can have smaller footprints despite the TSV's overheads [3], and lower power consumption thanks to the shorter wires [4].

Despite the aforementioned advantages, reliability has been a major concern of Through-Silicon-Vias due to their low yield rates [5, 6], vulnerability to thermal and stress, and the crosstalk issues of parallel TSVs [7–9]. In a 3D-DDR3 memory implementation [6], the statistics show that the defect rate of TSVs is nearly 0.63%. Defects on TSVs can occur in both random and cluster distributions [10] which create concerns about their fault-tolerance capabilities. Because of the natural parallel structure, TSVs also face the crosstalk challenge [11, 12]. Furthermore, the difference in thermal expansion coefficients of materials and temperature variations between two layers, which has been reported to reach up to 10°C [13], could lead to stress issues. To enhance the reliability of TSVs, there are three main approaches: (*i*) hardware fault-tolerance such as correction circuits [14], redundancies [10], reliability mapping [8]; (*ii*) information redundancy such as coding techniques [11, 15, 16] or re-transmission request [17]; or (*iii*) algorithm-based fault-tolerance [18–20]. Built-in-self-test (BIST) [21, 22] and external testing [23, 24] techniques are also proposed to help the system to determine whether a TSV has a defect.

Although numerous methods have been proposed to solve the reliability issues of TSVs, there are several problems that remain a challenge for designers. First, the redundancy-based method does not always support fault detection. Consequently, the system may require dedicated testing techniques. Even after correction, there is no guarantee that the recovered TSVs are healthy; so, on-line detection has become important for safety-critical applications. Second, a testing process using BIST [21, 22] or external testing [23, 24] usually cause interruptions of the system's operations and may lead to a considerable area cost and power consumption if the testing is performed in an on-chip and on-line manner. Third, besides simple coding techniques such as Parity, Hamming [15] or SECDED [16] (Single Error Correction, Double Error Detection), other coding techniques such as Reed-Solomon or BCH

are complicated making them unsuitable for high-frequency TSVs. On the other hand, the detection rates of SECDED or Hamming are low (one and two faults) which may lead to silent faults if multiple TSVs are failing. For instance, Hamming and SECDED can detect at most one and two faults, respectively. The exception is Orthogonal Latin Square Code (OLSC) [25] which provide low latency and modular design. However, OLSC does not provide extra detectability.

Because TSVs can operate at extremely high speeds, a simpler coding technique could be helpful for quickly correcting the occurred faults. Instead of detecting a limited number of faults, this coding technique should alert the system when multiple faults occur. This could help the system deciding how to perform the testing in order to understand the defect patterns or using algorithm-based methods to avoid the defected regions. Therefore, in this paper, we propose a new coding method named Two Dimensional Parity Product Code (2D-PPC) which is specially designed for correcting and detecting faults in TSV-based links. This work was presented in part at APPCAS 2019 [26]. The contributions of this paper are as follows:

- 2D Parity Product Code (2D-PPC) offers one-bit correction and at least two bits detection. With the same with of two dimension, 2D-PPC could be consider as an extended version of Orthogonal Latin Square code [25]. A Monte-Carlo simulation shows that 2D-PPC could detect an extremely higher number of bit-flips.
- Light-weight design of the proposed 2D-PPC's encoder and decoder. Design of 2D-PPC shows lower delay values than Hamming and SECDED. Even with 64 data bit-width, the delay sum of the encoder and decoder is 1.40 ns which is reasonably small. Moreover, the encoder has the ability to self-detect faults on its own circuit.
- The complexity and delay function of the encoding and decoding processes are presented. Here, the delay complexity is only $\mathcal{O}(\log_2(\sqrt{n}))$ while it is $\mathcal{O}(\log_2(n))$ for Hamming and SECDED (n is the input's bit-width).

The organization of this paper is as follows: Section 2 reviews the existing literature on coding techniques and TSV fault-tolerances. Section 3 presents the proposed 2D-PPC. Section 4 provides the evaluation environment and results. Finally, Section 5 concludes the paper.

2 Related Works

As previously mentioned, we can classify the TSV fault-tolerance into three main approaches: hardware-based, information redundancy, and algorithm-based. This section aims to briefly discuss these approaches in addition to the works conducted for TSV testing.

For the hardware-based fault-tolerance, there are three basic ideas: correction circuits [14], redundancy [10, 27] and reliability mapping [8, 20]. In [14], the authors presented a correction circuit for timing violation correction where long latency or highly dropped voltage TSVs are corrected using a dedicated circuit (a comparator for raising the voltage). Despite bringing several benefits, this technique is limited in terms of correctability. Using redundancies [27] to replace the failed ones entirely is also a common method. When a TSV is failed, the system maps its signals to a healthy spared TSV. Finally, reliability can be further enhanced with fault-tolerant mapping awareness. For instance, Ye *et al.* [8] use a mapping technique to put TSVs' positions during the layout process which can enhance the fault-tolerance technique.

Another fault-tolerance method is to use information redundancy. In other words, to be able to detect and correct faults, a code-word with redundant bits is used instead of the original data in the channels. Hamming code [15], which can detect and correct one faulty bit, is apparently the most important coding technique. SECDED by Hisao [16] is also extremely useful with the help of HARQ (Hybrid-Automatic Retransmission Request) mechanism. SECDED can detect two faults in a flit which could be re-transmitted for further correction as HARQ. In [28] and [29], authors present several variations of Hamming code using specified matrices which can correct two or even three adjacent fault bits. Thanks to their simple XOR functions, these codes are definitely simple and suitable for high-speed circuits; however, they have a limited number of detectable faults. In [26, 30–32], the authors have investigate the method to detect and localize multiple faults that overcome the limitation of ECCs. On the other hand, to tackle the cross-talk effect, Crosstalk Avoidance Code could be used [11]. Since using a dedicated coding technique seems inflexible, using an adaptive coding could be a suitable solution. In [17], packets are structured in 2D arrays and a Hamming code is used to correct a flit (column). When the decoder fails to correct the flit because of extra faulty bits, extra hamming codes for each index (row) are used. Therefore, the system can further correct faulty bits. Also, there are several powerful block coding methods such as Reed-Solomon [33] or Bose-Chaudhuri-Hocquenghem code [34] to help handle more faults; however, their calculations are too complicate which could lead to significant amount of area and power consumption.

When even hardware-based or information redundancy fail to correct the TSV failures, algorithm-based methods can help correct the communication at a higher level. For instance, fault-tolerant routing algorithms [18] could help 3D-NoCs work around faulty vertical links inside the network. Work in [20] presents a sharing algorithm method to adapt the network to the occurrences of cluster defects.

Besides fault-tolerance, fault-detection is also critical to help the system understand the faulty status. There are two in-field testing methods: Built-in-self-test (BIST) and external testing, in addition to two phases of manufacturing test: pre-bond and post-bond. In [21, 22], the authors presented other methods of TSV BIST for pin-hole and void defects. Probing before bonding with external testing [23, 24] is also helpful to improve the overall yield rate.

3 2D Parity Product Code

This section presents the proposed 2D Parity Product Code (2D-PPC). It is based on the Product-Code [26, 35, 36] approach and exploits the natural 2D array placement of TSVs. We first present the TSV organization and then the fault types are considered. The following parts demonstrate the encoding and decoding processes with equivalent circuits. Finally, we discuss the correctability and detectability of the coding technique.

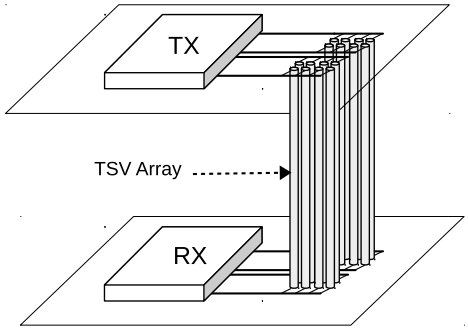

Figure 1. Inter-layer communication architecture: TX and RX stand for transmitter and receiver modules, respectively.

3.1 Fault Consideration

In this work, we mainly consider transient faults (soft error), open and short defects. Further impacts by crosstalk and stress issues could be detected and corrected if their behaviors match with the proposed fault model. Besides TSV's defects, faults on encoders and decoders are also considered in order to assess the system's overall reliability. The distribution of faults is defined as random.

3.1.1 Transient faults: Transient faults or soft errors are caused mainly by electromagnetic interference, cosmic rays [37], and alpha particles [38]. Notably, transient faults are reportedly occurred every 10^3 to 10^6 bits in aerospace applications [39]. This kind of faults is also increasingly affecting semiconductors as feature size is shrinking and operating voltages are reducing. Even the upper layers of the 3D-ICs act as natural shields from outside factors (i.e. cosmic rays), the faults are not entirely prevented.

3.1.2 Crosstalk effect: Since TSVs are placed in parallel between two adjacent layers, crosstalk has become a major effect. This effect is even more critical than 2D wires because a victim TSV could be affected by at most eight neighboring aggressors in 3D-ICs instead of two in 2D-ICs. Crosstalk may cause delays in voltage transition or even changing voltages without real driven transitions.

3.1.3 Permanent faults: There are two types of permanent defects: manufacturing defects and operating defects. Due to the imperfection during the manufacturing process, the permanent TSV defects are more frequent than other types of faults. TSV defects are usually of leakage (short), open (void), or bridge types [21, 40]. A TSV could be shortened to ground or Vdd which cause stuck-at faults. A bridge defect between two or more TSVs prevents them from transmitting different values at the same time. An open defect on a TSV increases its resistance which electrically disconnects its terminals or causes a transition delay. Aging, process variation or even temperature variation, which cause stress issues, could further increase the fault probabilities. Besides manufacturing defects, operating defects are also a considerable issue of TSV-based 3D-ICs. Due to the high temperature of 3D-ICs, other fault factors such as Electro-Migration, Time-Dependent-Dielectric-Breakdown, etc. are accelerated. Thermal Cycling is also another fault source due to the high difference in temperature between layers.

3.1.4 Fault modeling: Regarding behavior, we modeled the possible faults as stuck-at faults. For instance, the output logic value of a TSV is stuck to '0' or '1'. These behaviors are generally applied to soft errors as single event upset. The permanent defects could be physically modeled as RC models where the open and short resistances play important roles in their operations [27]. Delays caused by crosstalk and permanent faults could violate the timing constraints leading to sample the old values or metastability phenomenon could occur. This behavior is extremely hazardous for digital circuits and needs to be addressed appropriately using dedicated circuits [14, 41]. For a simple fault model, we use stuck-at faults for these type of faults.

3.2 TSV Organization

Assuming that a group of TSVs is organized in a 2D array of $M \times N$, as shown in Figure 1. Originally, a set of TSVs is organized as follows:

$$\text{TSVs} = \begin{bmatrix} T_{0,0} & T_{0,1} & \dots & T_{0,N-1} \\ T_{1,0} & T_{1,1} & \dots & T_{1,N-1} \\ \dots & \dots & \dots & \dots \\ T_{M-1,0} & T_{M-1,1} & \dots & T_{M-1,N-1} \end{bmatrix}, \quad (1)$$

where $T_{i,j}$ represents a TSV in the i^{th} row and the j^{th} column. As a product-code, for each row i and column j, we add an array of *row* parity-bits TSVs (CR_i) and an array of *column* parity-bits TSVs (CC_j). Then, there is an extra TSV CU for the *ultimate* check bits. The coded TSVs are as follows:

$$\text{Coded\_TSVs} = \begin{bmatrix} T_{0,0} & \dots & T_{0,N-1} & CR_0 \\ T_{1,0} & \dots & T_{1,N-1} & CR_1 \\ \dots & \dots & \dots & \dots \\ T_{M-1,0} & \dots & T_{M-1,N-1} & CR_{M-1} \\ CC_0 & \dots & CC_{N-1} & CU \end{bmatrix}. \quad (2)$$

Even when a group TSVs is not organized as a two dimensional array, we still can manage its data in a 2D array to apply the proposed technique. For instance, a group of 15 TSVs can be considered as a 4×4 group with one dummy value.

3.3 Encoding

For each transmission, a TSV $T_{i,j}$ sends a bit $b_{i,j}$, CR_i sends a row-parity bit r_i, CC_j sends a column-parity

bit c_j and CU sends an ultimate-parity bit u which is a member of a coded flit F:

$$F_k = \begin{bmatrix} b_{0,0} & b_{0,1} & \dots & b_{0,N-1} & r_0 \\ b_{1,0} & b_{1,1} & \dots & b_{1,N-1} & r_1 \\ \dots & \dots & \dots & \dots & \dots \\ b_{M-1,0} & b_{M-1,1} & \dots & b_{M,N-1} & r_{M-1} \\ c_0 & c_1 & \dots & c_{N-1} & u \end{bmatrix}, \quad (3)$$

where

$$\begin{aligned} r_i &= b_{i,0} \oplus b_{i,1} \oplus \cdots \oplus b_{i,N-1}, \\ c_j &= b_{0,j} \oplus b_{1,j} \oplus \cdots \oplus b_{M-1,j}, \\ ur &= r_0 \oplus r_1 \oplus \cdots \oplus r_{M-1}, \\ uc &= c_0 \oplus c_1 \oplus \cdots \oplus c_{N-1}, \\ u &= ur = uc = \oplus_{i=0}^{N-1} \oplus_{j=0}^{M-1} (b_{i,j}). \end{aligned} \quad (4)$$

Note that the symbol $\oplus$ stands for XOR function. This is also a self-detecting circuit where the bit u can be obtained by two separate equations (ur and uc). If there is a fault in their XOR functions, the two equations may give different ur and uc values. Therefore, the encoder can detect a failure by comparing:

$$\mathsf{Enc\_Error} = \begin{cases} 1, \text{ if } ur \neq uc, \\ 0, \text{ otherwise.} \end{cases} \quad (5)$$

If Enc_Error is equal to '1' in a short period of time, there is a transient fault. If this behavior continues for a long period or frequently occurs, a permanent fault can be detected. The self-detection ability is verified and discussed in Section 4.4.

The architecture of 2D-PPC encoders is shown in Figure 2 where the *Row Encoder*, *Col. Encoder*, and *Ulti. Encoder* are for encoding the rows, columns and ultimate bits, respectively. These encoders share the same parity encoder architecture (known as XOR-tree), as shown in Figure 3. The Enc_Error signal for informing the faulty status of the encoding process is obtained by comparing uc and ur. If designers do not desire to detect faults on the encoder, this signal can be simply removed to reduce the area cost (one XOR-tree and one XOR gate). The coding rate (CR) of 2D-PPC($N \times M$) with N rows and M columns is defined as:

$$CR = \frac{MN}{(M+1)(N+1)}. \quad (6)$$

The expected number of gates ($\mathbf{G}$) and expected delay (τ) of the encoding process is shown in Equation (7) and Equation (8), respectively.

$$\mathbf{G}^{\text{encoder}}_{\mathsf{XOR\_2X1}} = 2MN - 1 \quad (7)$$

$$\begin{aligned} \tau^{\text{encoder}}_{\mathsf{Dout}} &= \begin{cases} \tau_{\mathsf{XOR\_2X1}} \times (\text{ceil}(\log_2(\max(M,N))) \\ \quad +\text{ceil}(\log_2(M)))) \text{ if } u = ur \\ \tau_{\mathsf{XOR\_2X1}} \times (\text{ceil}(\log_2(\max(M,N))) \\ \quad +\text{ceil}(\log_2(N)))) \text{ if } u = uc \end{cases} \\ \tau^{\text{encoder}}_{\mathsf{Enc\_Error}} &= 2 \times \tau_{\mathsf{XOR\_2X1}} \times \text{ceil}(\log_2(\max(M,N))) \end{aligned} \quad (8)$$

From Equation (7) and Equation (8), with a given n bit ($M = N = \sqrt{n}$), the area cost and delay complexity

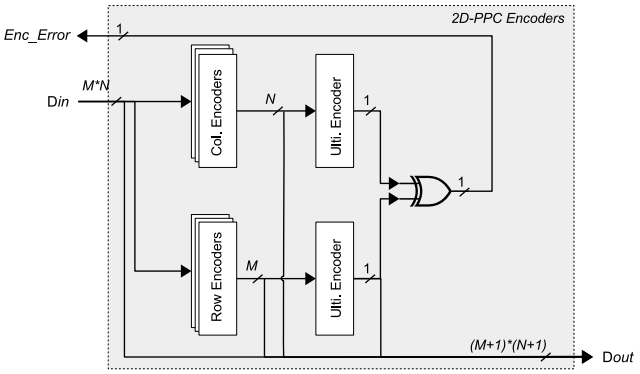

Figure 2. 2D-PPC Encoder Architecture.

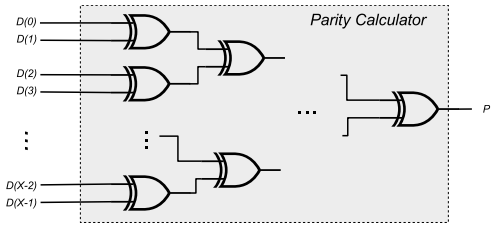

Figure 3. Parity architecture using XOR tree.

are $\mathcal{O}(n)$ and $\mathcal{O}(\log_2(\sqrt{n}))$, respectively. In comparison, Hamming's and SECDED's area cost and delay complexities are $\mathcal{O}(n)$ and $\mathcal{O}(\log_2(n))$. This means 2D-PPC provide better scalability in terms of delay.

3.4 Decoding

By using parity checking, the decoder can find the column and row indexes of the flipped bit. The parity equations are as follows:

$$\begin{aligned} sr_i &= b_{i,0} \oplus b_{i,1} \oplus \cdots \oplus b_{i,N-1} \oplus r_i, \\ sc_j &= b_{0,j} \oplus b_{1,j} \oplus \cdots \oplus b_{N-1,j} \oplus c_j, \\ sr_N &= r_0 \oplus r_1 \oplus \dots r_{M-1} \oplus u, \\ sc_M &= c_0 \oplus c_1 \oplus \dots c_{N-1} \oplus u. \end{aligned} \quad (9)$$

The outputs of Equation (9) are two arrays of parity column (sc) and parity row (sr). If there is one or no flipped bit, the decoder can correct it using a masked:

$$\mathsf{Mask} = \begin{bmatrix} m_{0,0} & \dots & m_{0,N-1} & m_{0,N} \\ m_{1,0} & \dots & m_{1,N-1} & m_{1,N} \\ \dots & \dots & \dots & \dots \\ m_{M-1,0} & \dots & m_{M-1,N-1} & m_{M-1,N} \\ m_{M,0} & \dots & m_{M,N-1} & m_{M,N} \end{bmatrix}, \quad (10)$$

where

$$m_{i,j} = \begin{cases} 1, \text{ if } sr_i == 1 \text{ and } sc_j == 1, \\ 0, \text{ otherwise.} \end{cases} \quad (11)$$

For each received flit $\hat{F}_k$, the corrected flit F_k is obtained by:

$$F_k = \hat{F}_k \oplus \mathsf{Mask}. \quad (12)$$

The decoder fails to correct when there are two or more faults. In this fashion, the decoder sends a NACK signal and a hybrid automatic retransmission request

(HARQ) is used to perform correction. To support HARQ, the decoder has to detect the occurrence of faults by summarizing the number of flipped bits in row and column as follows:

$$\begin{aligned} fr &= \sum_{i=0}^{N+1} sr_i, \\ fc &= \sum_{i=0}^{M+1} sc_i, \\ \mathsf{NACK} &= (fr \geq 2)\ \mathrm{OR}\ (fc \geq 2). \end{aligned} \tag{13}$$

Note that the above equations require adders and comparators which are probably over-complicated for high-speed coding techniques. To simplify the calculation of NACK, decoders can simply check either *sc* or *sr* if they are not all-zeros or one-hot values. For instance, with $M = 4$ and $N = 3$, fr, fc, and NACK can be expressed as:

$$\begin{aligned} fr = \neg\ &(\overline{sr_0}sr_1sr_2sr_3 + sr_0\overline{sr_1}sr_2sr_3 \\ &+ \overline{sr_0}sr_1\overline{sr_2}sr_3 + \overline{sr_0}sr_1sr_2\overline{sr_3} \\ &+ \overline{sr_0}sr_1\overline{sr_2}sr_3) \\ fc = \neg\ &(\overline{sc_0}sc_1\overline{sc_2} + sc_0\overline{sc_1}sc_2 + \overline{sc_0}sc_1\overline{sc_2} \\ &+ \overline{sc_0}\overline{sc_1}sc_2) \\ \mathsf{NACK} = fr + fc \end{aligned} \tag{14}$$

Figure 4 shows the architecture of the decoder. Similarly to the encoder, there are modules using XOR-trees (*Col. Decoder* and *Row Decoder*). Then, two arrays *sr* and *sc* are used for masking the faults. By taking the sum the number of faults in rows and columns ($\sum$ Row Faults and $\sum$ Col. Faults), the decoder can determine whether there are multiple faults occurrence. The NACK signal is used for retransmission using the HARQ protocol.

The expected number of gates (**G**) and delay (τ) of the decoding process are shown in Equation (15) and Equation (16), respectively. Note that the synthesizer could pick different gates with multiple inputs to optimize the area and timing.

$$\begin{aligned} \mathbf{G}^{\mathrm{decoder}}_{\mathsf{XOR\_2X1}} &= M(N+1) + N(M+1) + MN \\ \mathbf{G}^{\mathrm{decoder}}_{\mathsf{INV}} &= N + M + 2 \\ \mathbf{G}^{\mathrm{decoder}}_{\mathsf{AND\_2X1}} &= (N+2)N + (M+2)M \\ \mathbf{G}^{\mathrm{decoder}}_{\mathsf{OR\_2X1}} &= M + N \end{aligned} \tag{15}$$

$$\begin{aligned} \tau^{\mathrm{decoder}}_{\mathsf{Mask}} &= \tau_{\mathsf{XOR\_2X1}} \times \mathrm{ceil}(\log_2(\max(M+1, N+1))) \\ \tau^{\mathrm{decoder}}_{\mathsf{Dout}} &= \tau^{\mathrm{decoder}}_{\mathsf{M}} + \tau_{\mathsf{XOR\_2X1}} \\ \tau^{\mathrm{decoder}}_{\mathsf{Sum\_Faults}} &= \tau_{\mathsf{INV}} + \mathrm{ceil}(\log_2(\max(M+1, N+1)))) \\ &\quad \times (\tau_{\mathsf{AND\_2X1}} + \tau_{\mathsf{OR\_2X1}}) \\ \tau^{\mathrm{decoder}}_{\mathsf{NACK}} &= \tau^{\mathrm{decoder}}_{\mathsf{M}} + \tau^{\mathrm{decoder}}_{\mathsf{Sum\_Faults}} + \tau_{\mathsf{OR\_2X1}} \end{aligned} \tag{16}$$

From Equation (15) and Equation (16), with a given n bit ($M = N = \sqrt{n}$), the area cost and delay complexity are $\mathcal{O}(n)$ and $\mathcal{O}(\log_2(\sqrt{n}))$, respectively. In comparison, both of Hamming's and SECDED's area cost and delay complexities are $\mathcal{O}(n)$ and $\mathcal{O}(\log_2(n))$.

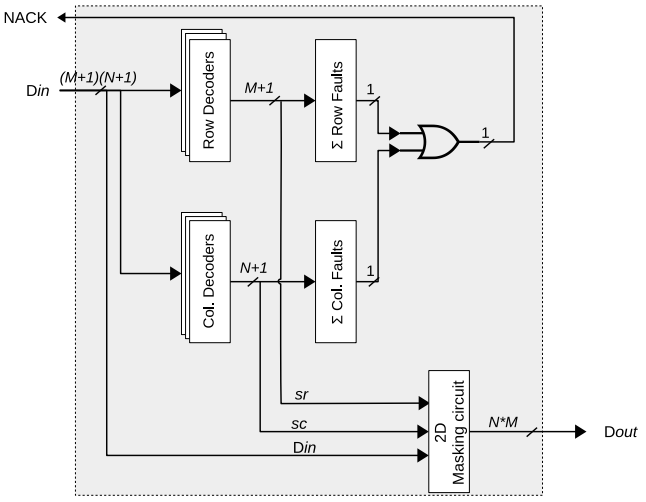

Figure 4. 2D-PPC Decoder Architecture.

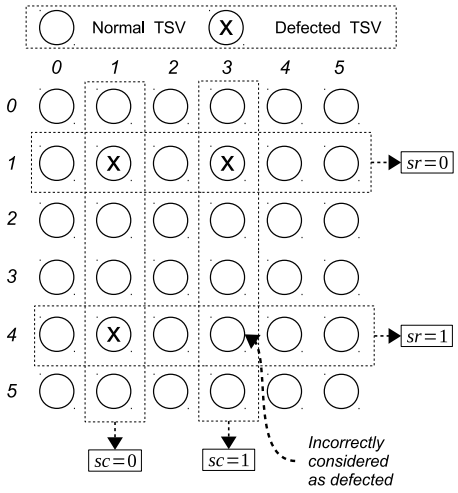

Figure 5. Undetectable pattern of 2D-PPC.

3.5 Correctability and Detectability

In general, 2D-PPC can ensure the ability to correct one and detect two flipped bits. However, if there are more than two flipped bits, 2D-PPC also has chances to detect them. For instance, three flipped bits with index (1,1), (2,3), and (0,4) of a 2D-PPC (6×6) results in a row check $sr = 000111$ and a column check $sc = 011010$. By determining that the *sr* and *sc* values have multiple bits '1' (Equation (13) or (14)), the decoder can detect more than two faults.

Although 2D-PPC can detect more than two faults, there is a weak point in its detection approach that always prevents it from detecting three faults. For instance, if bits with indexes (i, j), (i, k) and (l, j) are flipped, both cr_i and sc_j are '0' which make the decoder fails to detect while both cc_k and sr_l could be '1'. This syndrome makes the decoder understand that there is one fault and corrects the bit $b_{l,k}$. Figure 5 shows a simple illustration of such a case. In this case, a 2D-PPC (6×6) having three flipped bits with index (1,1), (1,3) and (4,1) is decoded to have $sc = 001000$ and $sr = 001000$. Because the flipped bits belong to the same row and column, the parity check bit (*sc* and

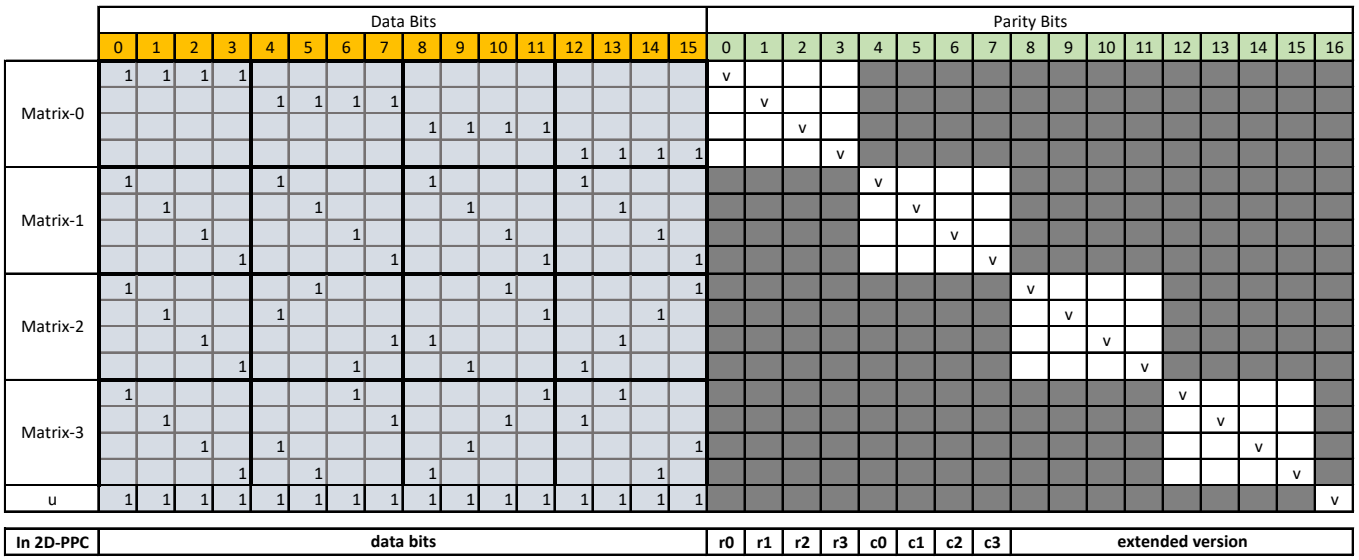

Figure 6. Extending 2D-PPC using orthogonal Latin square.

sr) is calculated as correct. However, because row 3 and column 3 are determined as having flipped bit, the decoder determines that bit (3,3) is flipped.

Despite this behavior, 2D-PPC still ensures the detection of at least two faults which is similar to SECDED and better than Hamming. If we consider that the fault distribution is stochastic, the probability of a TSV having a defect is $P_{f,TSV}$. The probability of having a failed set of three TSVs in the aforementioned case is: $P = P^3_{f,TSV}\frac{4MN}{(M+1)^2(N+1)^2}$. Since $P_{f,TSV} \ll 1$ and $\frac{4MN}{(M+1)^2(N+1)^2} < 1$, the probability of having the undetectable case is relatively small. The Monte-Carlo simulation of fault detection is presented in Section 4.1 to investigate the detection capacity of the coding technique.

3.6 Extending 2D-PPC using Orthogonal Latin Square

In order to correct more faults, we could extend 2D-PPC based on orthogonal Latin square. Note that it will limit the shape of 2D-PPC to square ($M = N$). In [42], the authors implement the low power OLSC using the orthogonal Latin square matrices for encoding and decoding. Obviously, squared 2D-PPC ($M = N$) without u bit is a case of OLSC. There are two extension using orthogonal Latin square: (1) to provide extra bit correction and (2) to break the undetectable pattern.

3.6.1 Correct extra bits: Therefore, by using additional orthogonal Latin square matrices, we could extend 2D-PPC to correct more faults.

Figure 6 shows the method to extend using orthogonal Latin square. The *Matrix-0* and *Matrix-1* is used in the baseline version of 2D-PPC where the parity bits are the result of calculating parity of columns and rows . To have different coding, *Matrix-2* and *Matrix-3* are used. In OLSC, to correct T faults ($T < M$), it requires $2T$ matrices and $2MT$ parity bits. Because of its modularity, applying this method does not affect the delay and area cost complexity. It also reserve the feature of self-detection by using u bit.

Although this method of extension could provide extra correct bits, the area overhead due to adding more TSVs is unreasonable.

3.6.2 Breaking the undetectable pattern: We also could observe that the design for *Matrix-2* and *Matrix-3* is identical to the original matrices of 2D-PPC. While the original matrix could be limited the undetectable, simply switching the different matrices could break this pattern. The extra cost and latency are only $M \times N$ multiplexers and a MUX 2:1 delay, respectively.

4 Evaluation

The 2D-PPC circuit is designed in Verilog-HDL with 45 nm process technology. The design is implemented using EDA tools by Synopsys. A software version of 2D-PPC is also implemented using Python. We first compare the 2D-PPC with other coding techniques in terms of coding rates and complexity function. Here, we choose SECDED and Hamming codes which are the two most well-known and well-used coding techniques. Then, the real implementation results are presented and compared. Also, we compare the energy per bit of the proposed design. The self-checking and self-correction ability of the encoder and decoder is presented later.

4.1 Coding Performance

Figure 7 summarizes the coding rates of 2D-PPC and compares them to Hamming and SECDED codes. Coding rate is the useful (or non-redundant) proportion of the codeword. 2D-PPC has lower coding rate while giving a similar ability as it can correct one fault and detect at least two faults. However, as we previously discussed, 2D-PPC can perform the detection of more than two flipped bits.

In order to study the detection ability of 2D-PPC, we perform a 10,000 cases Monte-Carlo simulation

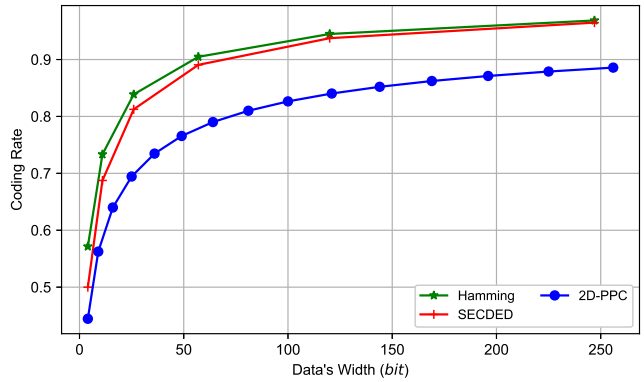

Figure 7. Coding rates of 2D-PPC in comparison to Hamming and SECDED.

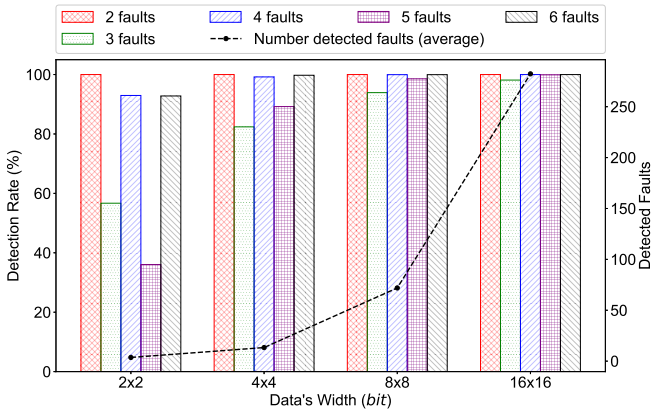

Figure 8. 2D-PPC detection ability evaluation.

represented in Figure 8. Monte-Carlo simulation is performed by randomizing the fault position in the channel and calculating the averaged value of the results. With 2D-PPC (2×2), the results show that the average number of detected faults is 3.6370 which is better than SECDED and Hamming code (2 and 1 faults, respectively). However, the in-depth analysis using Monte-Carlo simulation also points out that only 56.69% and 36.05% of 3-faults and 5-faults cases, respectively, were detected. This is due to the drawback of the 2D-PPC that it cannot detect the kind of pattern as shown in Figure 5. Even though, 2D-PPC provides excellent performance with a higher number of data's bit-width because there is less chance for the worst cases of 2D-PPC to happen. As we calculate, the probability of having an undetected pattern is $P = P_{f,TSV}^3 \frac{4MN}{(M+1)^2(N+1)^2}$. By having a larger number of data bit-width, both M and N increase. Therefore, the probability of undetected pattern is decreased. In fact, the results show that more than 99% of 4+ fault patterns have been detected. The three faults pattern detection rates of 4×4, 8×8 and 16×16 are 82.39%, 93.91% and 98.14%, respectively. In summary, the detection rate of 2D-PPC is extremely high, especially with higher data's bit-width.

4.2 Hardware Implementation

The hardware implementations of 2D-PPC are presented in Figure 9. For a fair comparison, the Enc_Error signal is optimized in the synthesizing process. This part will be separately evaluated in Section 4.4. Here, we compare 2D-PPC to SECDED and Hamming with the same data bit-width. We also add the results from [28] and [29] which provide two or three adjacent fault correction coding techniques. The results from [28] and [29] are presented in both area and delay optimization while our design simply targeted for timing optimization.

The results demonstrate that 2D-PPC provides several benefits over SECDED and Hamming. The complexities of 2D-PPC's encoders and decoders are lower than the other two. In particular, the area cost of the encoder and decoder for 64-bit of 2D-PPC are 17.01% and 15.95% less than the SECDED's ones. The latency of 2D-PPC encoders and decoders are also smaller thanks to the narrower XOR trees. For 64-bit, they are 22.67% and 49.38% lower than the SECDED's ones. In comparison to the best area optimized (AO) results in [28] and [29], and despite the fact that we use more parity bits, the proposed design (encoder and decoder) only incurs 15.96% and 14.20% extra area cost, respectively. The best delay optimized (DO) design in [28] and [29] reduces the latency by 67.14% and 64.29% when compared to 2D-PPC; however, their complexities are 8 times higher.

Detailed results of 64 data bit-width implementations are shown in Table I. Besides the works in [28] and [29], we also perform the comparison with results obtained from [17] which are implemented in 65 nm technology. Even when scaling to 45 nm, the area cost of Hamming Product Code (HP-HARQ-II) in [17] is 8.11 times higher than 2D-PPC. The BCH [17] code provides multi-bits correction; however, its complexity is 50 times more than the proposed one. HP-HARQ-II's and BCH's latencies are 28.57% and 18.57% lower despite using older technology. However, our latency is still extremely low (0.58 ns and 0.82 ns). With the delay complexity $\mathcal{O}(\log_2(\sqrt{n}))$, the results are expected to be lower with higher data-widths. Meanwhile, the area cost is similar to Hamming and SECDED which are two simple coding techniques. It is important to mention that the area cost results have not taken into account the area of TSV. As previously shown in the coding rate evaluation in Figure 7, our design demands more additional TSVs than the others. With the same 64 data bit-width, 2D-PPC uses 81 code-word bit-width (or TSVs) while Hamming, SECDED, BCH use 71, 72 and 85 code-word bit-width (or TSVs), respectively.

4.3 Energy Evaluation

To understand the energy consumed by the proposed 2D-PPC, we investigate a pair of 8×8 encoder and decoder. Furthermore, we compare the results with SECDED (64,72) and Hamming (64,71). Here, we perform the energy per bit evaluation for several test cases: fault-free, 1-fault, and 2-faults. With the 2-faults case, HARQ is added. Note that the energy consumption is only calculated for the encoder and decoder, the wire and the register for ARQ is omitted to keep a fair comparison. Table II represents the energy evaluation. We

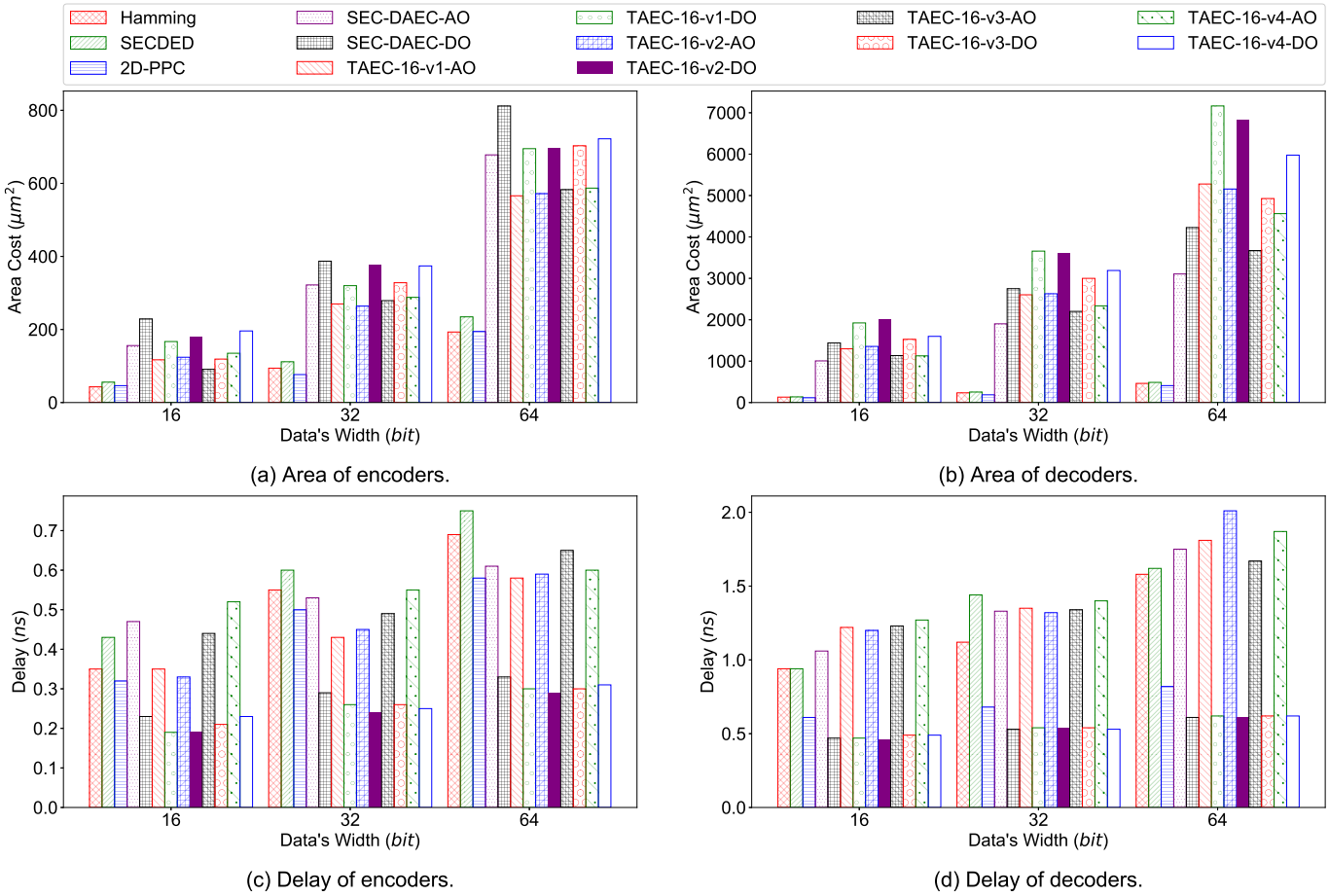

Figure 9. Hardware implementation results.

Table I
HARDWARE IMPLEMENTATION RESULTS: "AO" AND "DO" ARE AREA OPTIMIZATION AND DELAY OPTIMIZATION, RESPECTIVELY. SCALING FROM 65 *nm* TO 45 *nm* USES EQUATIONS ON [43]

Scheme	**Tech. (nm)**	**k** (bit)	**n** (bit)	**Area Cost (μm^2)**				**Latency (ns)**			
				Encoder		Decoder		Encoder		Decoder	
				AO	DO	AO	DO	AO	DO	AO	DO
Hamming [15]	45	64	71	193.1200		463.1060		0.69		1.58	
SECDED [25]	45	64	72	234.6120		487.0460		0.75		1.62	
HP + HARQ-II [17]	65	64	72	9792.5				0.41		0.59	
	45 (*scaled*)	64	72	4896.25				0.29		0.42	
ARQ (CRC-5) [17]	65	64	69	3605.6				0.37		0.41	
	45 (*scaled*)	64	69	1802.8				0.26		0.29	
BCH [17]	65	64	85	77353.2				0.42		0.72	
	45 (*scaled*)	64	85	38676.6				0.30		0.51	
SEC-DAEC [28]	45	64	72	678	812	3106	4227	0.61	0.33	1.75	0.61
TAEC-64-v1 [29]	45	64	72	566	695	5279	7165	0.58	0.30	1.81	0.62
TAEC-64-v2 [29]	45	64	72	572	696	5158	6833	0.59	0.29	2.01	0.61
TAEC-64-v3 [29]	45	64	71	583	703	3672	4928	0.65	0.30	1.67	0.62
TAEC-64-v4 [29]	45	64	71	587	722	4563	5976	0.60	0.31	1.87	0.62
2D-PPC(8×8)	45	64	81	194.7120		409.3740		0.58		0.82	

Table II
Energy Evaluation with NANGATE 45 *nm*, 250 *MHz*, Random 64 Data-Bit

Scheme	Test Case	Energy per data bit (J/bit)	
		Encoder	Decoder
Hamming(64,71)	free	1.0720e-13	3.3036e-13
	one fault	1.0720e-13	3.5481e-13
SECDED(64,72)	free	1.2788e-13	3.2535e-13
	one fault	1.2788e-13	3.7926e-13
	two faults	1.2788e-13	3.0654e-13
2D-PPC(64,81)	free	8.5882e-14	5.0275e-13
	one fault	8.5882e-14	5.0526e-13
	two faults	8.5882e-14	5.0965e-13

Table III
Brute-Force Fault Detection Simulation Results of 2D-PPC(8×8)'s Encoder and Decoder

	Cases	Encoder	Decoder
Gates		127	307
Detection signal		Enc_Error	NACK
Faults	Inserted	127 (100%)	307 (100%)
	Self-corrected	-	116 (37.78%)
	Detected	127 (100%)	12 (3.91%)
	Undetected	0	179 (58.31%)

perform the encoding and decoding of a randomized 64-bit data. The number of flits is 1000 and the clock frequency is 250 MHz. The power estimation is done using Synopsys PrimeTime.

Thanks to the simplicity of the encoding process, our decoder consumes less energy per bit than SECDED and Hamming codes (-32.84% and -19.86%). However, because of having more codeword-bit (81-bit instead of 71-bit and 72-bit), our decoding process demands more energy. Without fault, our decoder consumes 52.94% and 55.30% more energy than Hamming's and SECDED's decoders. In summary, our encoding and decoding processes together consume 28.90% and 17.43% additional energy when compared to Hamming and SECDED.

4.4 Self-Checking Encoders/Decoder

In this section, we evaluate the ability to self-check the correctness of encoders and decoders. Here, we insert faults in the netlist file using Python and Regular Expression. For each gate in the netlist, we attach it with an error injector which can toggle or prefix the output value of the gate to create stuck-at faults or single event upset. Then, a controller is used to select which gate is injected. The post-synthesized netlist is processed using Python and the Regular Expression library to allow inserting faults in each gate of the design. Instead of using a Monte-Carlo simulation as in [44], due to the small number of gates in both encoder and decoder, we use the brute-force simulation to find out the detection coverage. In order to test the correctness of the encoder and decoder, we insert 1000 flits and monitor whether it can correctly detect the fault. Here, we use the SECDED (8×8). As previously mentioned, the design in this section is implemented separately. The Enc_Error signal is optimized during the synthesis process since it is always '0'. By conservatively keeping this signal in Design Compiler, we can perform the self-checking process. This encoder uses 127 gates instead 122 gates in the optimized version.

As shown in Table III, the encoder can detect all self-inserted faults. Because the u bit is taken from either uc or ur, faults on unselected branch do not corrupt the codewords. The decoder can self-correct 116 out of 306 faults thanks to the correctability of the decoder. The further impact of 12 other faults can be detected by the decoder which sends out NACK. Lastly, there are 178 out of 307 faults (58.31%) that have have led to corruptions without being corrected or detected.

In summary, the encoder can completely detect any single fault inside itself. The decoder can self-correct and self-detect around 40% of single faults. Since the stress caused by the TSV implementation could be critical and contribute to the increase in fault probability of any circuit, the high reliability of the encoder and decoder is extremely important.

5 Conclusion

This paper presents the 2D Parity Product Code (2D-PPC) to enhance the reliability of TSV-based 3D-IC designs. By exploiting the inherent 2D array organization of TSVs, the proposed approach can efficiently represent the fault manifestation in TSV-based systems allowing it to correct one and detect at least two faults in a set of TSVs. In addition, 2D-PPC was designed to be self-aware and was capable to detect possible fault occurrences in the router's encoders/decoders.

From the conducted experiments, and in contrast to conventional coding schemes that are limited to detecting two faults at most, the proposed 2D-PPC has demonstrated its ability to detect over 71 faults on average, for a 64 data bit-width case. Our analysis also showed that the delay complexity of 2D-PPC is $\mathcal{O}(\log_2(\sqrt{n}))$ which is significantly lower than that of Hamming/SECDED ($\mathcal{O}(\log_2(n))$). Furthermore, the 2D-PPC's encoder and decoder reduce the area cost by 17.01% and 15.95%, and decrease the latency by 22.67% and 49.38% when compared to the SECDED ones, respectively.

As a future work, we plan to apply the 2D-PPC to a dedicated 3D-ICs architecture (e.g., 3D-RAM, 3D-NoCs) to investigate the impact on the overall system. Extending the technique with adaptive coding and different based coding methods is another possible direction.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.312.

References

[1] J. Cho, E. Song, K. Yoon, J. S. Pak, J. Kim, W. Lee, T. Song, K. Kim, J. Lee, H. Lee *et al.*, "Modeling and analysis of through-silicon via (TSV) noise coupling and suppression using a guard ring," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 220–233, 2011.

[2] J. Kim, J. S. Pak, J. Cho, E. Song, J. Cho, H. Kim, T. Song, J. Lee, H. Lee, K. Park *et al.*, "High-frequency scalable electrical model and analysis of a through silicon via (TSV)," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 181–195, 2011.

[3] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2009, pp. 234–241.

[4] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs: The pros and cons of going vertical," *IEEE Design & Test of Computers*, vol. 22, no. 6, pp. 498–510, 2005.

[5] J. U. Knickerbocker, P. S. Andry, B. Dang, R. R. Horton, M. J. Interrante, C. S. Patel, R. J. Polastre, K. Sakuma, R. Sirdeshmukh, E. J. Sprogis *et al.*, "Three-dimensional silicon integration," *IBM Journal of Research and Development*, vol. 52, no. 6, pp. 553–569, 2008.

[6] U. Kang, H.-J. Chung, S. Heo, D.-H. Park, H. Lee, J. H. Kim, S.-H. Ahn, S.-H. Cha, J. Ahn, D. Kwon *et al.*, "8 Gb 3-D DDR3 DRAM using through-silicon-via technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 111–119, 2010.

[7] G. Van der Plas, P. Limaye, I. Loi, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Linten, M. Stucchi, G. Katti *et al.*, "Design issues and considerations for low-cost 3-D TSV IC technology," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 293–307, 2011.

[8] F. Ye and K. Chakrabarty, "TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation," in *Proceedings of the Design Automation Conference*. ACM, 2012, pp. 1024–1030.

[9] K. N. Dang and A. B. Abdallah, "Architecture and Design Methodology for Highly-Reliable TSV-NoC Systems," in *Horizons in Computer Science Research*. Nova Science Publishers, 2018, vol. 16, pp. 199–246.

[10] L. Jiang, Q. Xu, and B. Eklow, "On effective through-silicon via repair for 3-D-stacked ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 4, pp. 559–571, 2013.

[11] R. Kumar and S. P. Khatri, "Crosstalk avoidance codes for 3D VLSI," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2013, pp. 1673–1678.

[12] A. Eghbal, P. M. Yaghini, N. Bagherzadeh, and M. Khayambashi, "Analytical fault tolerance assessment and metrics for TSV-based 3D network-on-chip," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3591–3604, 2015.

[13] Y. J. Park, M. Zeng, B.-s. Lee, J.-A. Lee, S. G. Kang, and C. H. Kim, "Thermal analysis for 3D multi-core processors with dynamic frequency scaling," in *Proceedings of the IEEE/ACIS 9th International Conference on Computer and Information Science (ICIS)*, 2010, pp. 69–74.

[14] M. Cho, C. Liu, D. H. Kim, S. K. Lim, and S. Mukhopadhyay, "Design method and test structure to characterize and repair TSV defect induced signal degradation in 3D system," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2010, pp. 694–697.

[15] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[16] M.-Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395–401, 1970.

[17] B. Fu and P. Ampadu, "On hamming product codes with type-ii hybrid ARQ for on-chip interconnects," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 2042–2054, 2009.

[18] A. B. Ahmed and A. B. Abdallah, "Architecture and design of high-throughput, low-latency, and fault-tolerant routing algorithm for 3D-network-on-chip (3D-NoC)," *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1507–1532, 2013.

[19] ——, "Adaptive fault-tolerant architecture and routing algorithm for reliable many-core 3D-NoC systems," *Journal of Parallel and Distributed Computing*, vol. 93, pp. 30–43, 2016.

[20] K. N. Dang, A. B. Ahmed, Y. Okuyama, and A. B. Abdallah, "Scalable design methodology and online algorithm for TSV-cluster defects recovery in highly reliable 3D-NoC systems," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[21] Y. Lou, Z. Yan, F. Zhang, and P. D. Franzon, "Comparing through-silicon-via (TSV) void/pinhole defect self-test methods," *Journal of Electronic Testing*, vol. 28, no. 1, pp. 27–38, 2012.

[22] M. Tsai, A. Klooz, A. Leonard, J. Appel, and P. Franzon, "Through silicon via (TSV) defect/pinhole self test circuit for 3D-IC," in *Proceedings of the IEEE International Conference on 3D System Integration*, 2009, pp. 1–8.

[23] B. Noia and K. Chakrabarty, "Pre-bond probing of TSVs in 3D stacked ICs," in *Proceedings of the IEEE International Test Conference*, 2011, pp. 1–10.

[24] P.-Y. Chen, C.-W. Wu, and D.-M. Kwai, "On-chip TSV testing for 3D IC before bonding using sense amplification," in *Proceedings of the Asian Test Symposium*. IEEE, 2009, pp. 450–455.

[25] M. Y. Hsiao, D. C. Bossen, and R. T. Chien, "Orthogonal latin square codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 390–394, 1970.

[26] K. N. Dang, M. C. Meyer, A. B. Ahmed, A. B. Abdallah, and X.-T. Tran, "2D-PPC: A single-correction multiple-detection method for through-silicon-via faults," in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems*, 2019, pp. 109–112.

[27] Y. Zhao, S. Khursheed, and B. M. Al-Hashimi, "Online Fault Tolerance Technique for TSV-Based 3-D-IC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 8, pp. 1567–1571, 2014.

[28] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in *Proceedings of the 25th IEEE VLSI Test Symposium (VTS'07)*. IEEE, 2007, pp. 349–354.

[29] L.-J. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli, and J. A. Maestro, "MCU tolerance in SRAMs through low-redundancy triple adjacent error correction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 10, pp. 2332–2336, 2015.

[30] K. N. Dang, A. B. Ahmed, and X. T. Tran, "An on-communication multiple-TSV defects detection and localization for real-time 3D-ICs," in *Proceedings of the IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, 2019, pp. 223–228.

[31] K. N. Dang, A. B. Ahmed, A. B. Abdallah, and X.-T. Tran, "TSV-OCT: A Scalable Online Multiple-TSV Defects Localization for Real-Time 3-D-IC systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019.

[32] K. N. Dang, A. B. Ahmed, B. A. Abderrazak, and X.-T. Tran, "TSV-IaS: Analytic Analysis and Low-Cost Non-Preemptive on-Line Detection and Correction Method for TSV Defects," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, 2019, pp. 501–506.

[33] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[34] I. S. Reed and X. Chen, *Error-control coding for data*

networks. Springer Science & Business Media, 2012, vol. 508.
[35] R. M. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 8, pp. 1003–1010, 1998.
[36] F. Chiaraluce and R. Garello, "Extended Hamming product codes analytical performance evaluation for low error rate applications," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2353–2361, 2004.
[37] J. F. Ziegler and W. A. Lanford, "Effect of cosmic rays on computer memories," *Science*, vol. 206, no. 4420, pp. 776–788, 1979.
[38] T. C. May and M. H. Woods, "A new physical mechanism for soft errors in dynamic memories," in *Proceedings of the 16th International Reliability Physics Symposium*, 1978, pp. 33–40.
[39] J. Sosnowski, "Transient fault tolerance in digital systems," *IEEE Micro*, vol. 14, no. 1, pp. 24–35, 1994.
[40] K. Chakrabarty, S. Deutsch, H. Thapliyal, and F. Ye, "TSV defects and TSV-induced circuit failures: The third dimension in test and design-for-test," in *Proceedings of the IEEE International Reliability Physics Symposium*, 2012, pp. 5F–1.
[41] K. A. Bowman, J. W. Tschanz, N. S. Kim, J. C. Lee, C. B. Wilkerson, S.-L. L. Lu, T. Karnik, and V. K. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, 2009.
[42] S. E. Lee, Y. S. Yang, G. S. Choi, W. Wu, and R. Iyer, "Low-power, resilient interconnection with orthogonal latin squares," *IEEE Design & Test of Computers*, vol. 28, no. 2, pp. 30–39, 2011.
[43] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74–81, 2017.
[44] K. N. Dang, A. B. Ahmed, X.-T. Tran, Y. Okuyama, and A. B. Abdallah, "A comprehensive reliability assessment of fault-resilient network-on-chip using analytical model," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 11, pp. 3099–3112, Nov 2017.

Khanh N. Dang is currently an assistant professor at VNU Key Laboratory for Smart Integrated Systems, Vietnam National University Hanoi (VNU), Hanoi Vietnam. He received his B.Sc., M.Sc., and Ph.D. degree from VNU University of Engineering and Technology, University of Paris-Sud XI, and The University of Aizu, Japan in 2011, 2014, and 2017, respectively. Dr. Khanh N. Dang was visiting researcher at University of Aizu in 2019. His research interests include System-on-Chips/Network-on-Chips, 3D-ICs, and fault-tolerant systems.

Michael Meyer is currently an Assistant Professor at Waseda University. He was previously a post-doctoral researcher at the University of Aizu in Fukushima, Japan, as a member of the Data Networking Laboratory. He graduated from Rose-Hulman Institute of Technology, in Indiana, USA, with a B.S. in Computer Engineering in 2012, and then with a M.A. in Engineering Management in 2013. In 2017 he received a Ph.D. in Comp. Sci. and Eng. from the University of Aizu. He has worked for Texas Instruments before starting his Ph.D., and before that he had worked for Syntheon developing biomedical devices. His research interests cover on and off chip networks, reliability and photonics.

Akram Ben Ahmed received his M.S.E. and Ph.D. degrees in Computer Science and Engineering from the University of Aizu, Japan, in 2012 and 2015, respectively. He is currently with National Institute of Advanced Industrial Science and Technology (AIST), Japan. He was a postdoctoral researcher in the Department of Information and Computer Science, Keio University, Japan from 2014 to 2019. His current research interests include on-chip interconnection networks, reliable and fault-tolerant systems, and ultra-low-power embedded real-time systems.

Abderazek Ben Abdallah Abderazek Ben Abdallah is a full Professor of Computer Science and Engineering and the Head of the Division of Computer Engineering, the University of Aizu. He has been a faculty member at the University of Aizu since 2007. Before joining the University of Aizu, he was a research associate at the Graduate School of Information Systems, the Univ. of Electro-Communications at Tokyo from 2002 to 2007. He received the h.D. degree in computer engineering from the Univ. of Electro-Communications at Tokyo in 2002. His research falls primarily in the area of computer system and architecture, with an emphasis on adaptive/self-organizing systems, networks-on-chip/SoCs, processor micro-architecture and power & reliability-aware architectures. He is also interested in neuro-inspired systems and VLSI design for 3D-ICs. He has authored three books, published more than 150 journal articles and conference papers in these areas and given invited talks as well as courses at several universities. He has been a PI or CoPI of several projects for developing next generation high-performance reliable computing systems for applications in general purpose and pervasive computing. He is a senior member of IEEE and ACM and a member of IEICE.

Xuan-Tu Tran received a Ph.D. degree in 2008 from Grenoble INP (at the CEA-LETI), France, in Micro Nano Electronics. He is currently an associate professor at VNU University of Engineering and Technology, Vietnam National University, Hanoi (VNU). He was an invited professor at the University Paris-Sud 11, France (2009, 2010, and 2015), University of Electro-Communication, Tokyo (2019), Grenoble INP (2011, 2020), and adjunct professor at University of Technology Sydney (2017-2020). He is currently Director for the VNU Key Laboratory for Smart Integrated Systems (SISLAB). His research interests include design and test of systems-on-chips, networks-on-chips, design-for-testability, asynchronous/synchronous VLSI design, low power techniques, and hardware architectures for multimedia applications. He has published more than 80 journal articles and conference papers in these areas and given invited talks as well as courses at several universities.

He is a Senior Member of the IEEE, IEEE Circuits and Systems (CAS), IEEE Solid-State Circuits and Systems (SSCS), member of IEICE, and the Executive Board of the Radio Electronics Association of Vietnam (REV). He serves as Chairman of IEICE Vietnam Section, Chairman of IEEE SSCS Vietnam Chapter.